KERETA KAWALAN PINTAR

SMARTPHONE CONTROL CAR

AHMAD MIRZA BIN AZHAR E20181022729

REPORT SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR

DIPLOMA IN COMPUTER SCIENCE (INTERNET COMPUTING)

FACULTY ART, COMPUTING AND CREATIVE INDUSTRIES UNIVERSITI PENDIDIKAN SULTAN IDRIS

2020

# SUPERVISOR’S APPROVAL

**The Development of Smartphone Control Car**

By

**AHMAD MIRZA BIN AZHAR**

## E20181022729

This report was prepared under the supervision of the project supervisor, Dr Ismail @ Ismail Yusuf Panessai. It was submitted to the Faculty of Art, Computing and Creative Industries and was accepted in partial fulfillment of the requirements for the Diploma in Computer Science (Internet Computing).

Approved by

………………………………………….

Dr Ismail @ Ismail Yusuf Panessai

Project Supervisor

2020

## **ACKNOWLEDGEMENTS**

Alhamdullilah and praises to Allah S.W.T because of His almighty and His blessings, I was able to finish this research within the time duration given. Firstly, I would like to thank to my supervisor, Dr. Ismail @ Ismail Yusuf Panessai for the patient guidance, encouragement and advice he has provided throughout my time as his student. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly. Secondly, special appreciation also goes to my beloved parents Encik. Azhar Bin A. Samah and Puan Asiah Binti Hassan for their never ending support and constant encouragement throughout my study. I would also like to express my gratitude and indebtedness to them for always pray for my success and support me. Last but not least, I would like to give my gratitude to my dearest friend of Universiti Pendidikan Sultan Idris’s Diploma Computer Science (Internet Computing) students for their endless motivation, support and helped me with the development of the project. I could not have done this without you guys.

## **ABSTRACT**

Nowadays, smartphones are becoming more powerful with reinforced processors, larger storage capacities, richer entertainment function and more communication methods. Bluetooth is mainly used for data exchange; add new features to smart phones. ESP8266 WIFI technology and other similar techniques, with dramatic increase in Smart phone users, smart phones have gradually turned into an all-purpose portable device and provided people for their daily use. The work behind our Paper is to provide easier robot’s hardware architecture but with efficient computational platforms so that robot’s designer can focus on their research and tests instead of ESP8226WIFI connection infrastructure with combination of nodeMCUV2 motor shield. We present the implementation of our approach, where data collection is implemented as an Android mobile application and further analysis of the collected data is done in order to Control the vehicle and we can analyze the all control like Vibrations then any obstacles finding at the time we can easily to control our vehicles. The paper is designed to develop robot using ESP8266WIFI technology for remote operation and control by using BLYNK apps. The robotic vehicle will execute commands by BLYNK apps on mobile transmitted to the robot. Then Vibrations occur at the time indicate then we controlling easily then Airbag system will open same time we can avoiding Accidents. Then Ultrasonic sensor is used easily for obstacle detecting and Indicating easily. We can easily get the overall Vehicle Controls via BLYNK app Using Android Smart phones.

***Keywords:*** Smartphone Car Control

## **ABSTRAK**

Pada masa kini, telefon pintar semakin kuat dengan pemproses yang diperkuat, kapasiti penyimpanan yang lebih besar, fungsi hiburan yang lebih kaya dan lebih banyak kaedah komunikasi. Bluetooth digunakan terutamanya untuk pertukaran data; tambah ciri baru untuk telefon pintar. Teknologi WIFI ESP8266 dan teknik lain yang serupa, dengan peningkatan dramatik dalam pengguna telefon pintar, telefon pintar secara beransur-ansur berubah menjadi peranti mudah alih serba guna dan menyediakan orang untuk penggunaan harian mereka. Kerja di belakang projek kami adalah untuk menyediakan senibina perkakasan yang lebih mudah tetapi dengan platform pengkomputeran yang cekap supaya pencipta robot boleh memfokuskan pada penyelidikan dan ujian mereka dan bukannya infrastruktur sambungan ESP8226WIFI dengan gabungan perisai motor nodeMCUV2. Kami membentangkan pelaksanaan pendekatan kami, di mana pengumpulan data dilaksanakan sebagai aplikasi mudah alih Android dan analisa lanjut mengenai data yang dikumpulkan dilakukan untuk mengendalikan kenderaan dan kami dapat menganalisis semua kawalan seperti Getaran maka setiap halangan yang ditemui pada saat kami mudah untuk mengawal kenderaan kami. Kertas ini direka bentuk untuk membangunkan robot menggunakan teknologi ESP8266WIFI untuk operasi dan kawalan jarak jauh dengan menggunakan aplikasi BLYNK. Kenderaan robot akan melaksanakan arahan oleh aplikasi BLYNK pada peranti mudah alih yang dihantar ke robot. Kemudian Getaran berlaku pada waktu yang ditunjukkan maka kita mengawal dengan mudah maka sistem Airbag akan membuka waktu yang sama kita dapat menghindari Kemalangan. Kemudian sensor ultrasonik digunakan dengan mudah untuk mengesan halangan dan Menunjukkan dengan mudah. Kita boleh dengan mudah mendapatkan Kawalan Kenderaan secara keseluruhan melalui aplikasi BLYNK Menggunakan telefon Android pintar.

***Keywords:*** Kereta Kawalan Pintar

|  |
| --- |
|  |
|  |
|  |
|  |

# **CHAPTER 1**

# **INTRODUCTION**

This chapter will be discussed about the research of project development precisely. Research that been included are background study, problem statement, research aim, research objective, research significance, research question and research scope. Each of this sub-chapter will be explained so that it will be cleared on why the developer want to build Smartphone Control Car

## **1.1 BACKGROUND OF STUDY**

Using a Smartphone as the “brain” of a robot is already an active research field with several open opportunities and promising possibilities. In this project we present a review of current robots controlled by mobile phone and discuss a closed loop control systems using audio channels of mobile devices, such as phones and tablet computers. In our work, move the robot upward, backward, left and right side by the android applications such as Arduino, RC Car.

In recently Systems a robot is usually an electro-mechanical machine that is guided by computer and electronic programming. Many robots have been built for manufacturing purpose and can be available in industries around the world. Designing of the latest inverted robot which can be controlling using an APP for android mobile. The developed remote buttons in the android app is helpful so that we can control the robot motion within that we use Bluetooth communication to interface the microcontroller and android mobile. Controller can be interfaced to the Bluetooth module though UART protocol. From the commands received from android the robot motion can be controlled in upward, downward, front and back movements. The consistent output of a robotic system along with quality and repeatability are unmatched. Pick and Place robots can be re programmable and tools can be interchanged to provide for multiple applications. The disadvantages of the Existing system are its Complexity, unable to achieve low power consumption, Poor signal quality and Low Accuracy.

The Internet of Things (IOT) is the concept of connecting any device to the internet and to other connected devices. The IOT is a network of connected things and people all of which collect and share data about the way they are used and about the environment around them. Furthermore, devices and objects with built in sensors are connected to an Internet of Things platform, which integrates data from the different devices and applies analytics to share the most valuable information with applications built to address specific needs. The Internet of Things (IOT) has evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commodity sensor, and embedded systems.

IOT Smartphone Control Car is an ESP8266WIFI and NodeMCUV2 motor shield and IOT based system and will show users that mobile phone controlled RC Car movements by connecting to Wi-Fi. The project is designed to show the RC Car movement is controlled over the internet using a mobile phone and sends a message showing location of car and the movement of RC car are connecting with the internet. The purpose of this project is to make it easier for users to control RC car and find out how to know the location of RC car and the notification sending through the apps on Android mobile smartphone.

## **1.2 PROBLEM STATEMENT**

There are many problems that occur in the previous type of smartphone RC car. The motor shield are no detecting directly to the remote due by using Bluetooth remote. As the result the gear motor produce by this device is not the maximum it should moving. The better solution for this system to get the ESPWIFI and Motor Shield is smartphone control RC system. This is the main reason the project smartphone RC car is made. The ESP8266WIFI will sending the notifications from BLYNK App to get output and start to control using smartphone. Indirectly it will reduce the cost of buying battery’s remote. These systems also reduce the time for users to change the position of solar panel to face the sun.

## **1.3 RESEARCH AIM**

1. To investigate any current project for smartphone control car (IoT).
2. To design smartphone control car based on IoT.
3. To develop smartphone control car based on IoT.
4. To evaluate the performance of smartphone control car based on IoT.

## **1.4 RESEARCH OBJECTIVES**

The objective of this projects are:

1. What is the latest current project based on Iot?
2. What is the suitable of smartphone car control based on IoT?
3. How to develop smartphone control car based on IoT?
4. What is the propose of performance for smartphone car control based on IoT?

## **1.5 RESEARCH SCOPE**

This project will be focused on the control of RC car movement and the messages sent via mobile phones over the internet. The RC car project is to control the car by using smartphone and send a message on BLYNK to detect location of car and send the notifications using the internet. This RC car movement will be controlled using the mobile phone over the internet to facilitate RC car anywhere in a long distance.

## **1.6 CONCLUSION**

In conclusion, based on chapter 1 explain why this project was developed. The project was developed to control a movement car through the smartphone and send the notifications by using the internet. The project movement can be controlled in BLYNK application only. The problem statement will help the development of the project for fulfil the requirement so that the product of it will become more quality.

## **1.7 RESEARCH SIGNIFICANCE**

There are several research significances of this project which are:

1. This project is used to control the car through the smartphone by using BLYNK.
2. This project will be controlled via smartphone, so there is no problem that happened with wired one.

# **CHAPTER 2**

## **LITERATURE REVIEW**

This chapter will give various type of information that focused on the project that been proposed. With this chapter, it helps to guide in development of the project. For the proposed project which is Smartphone RC car, this chapter will discuss Introduction of Internet of Thing and devices that be used in Internet of Thing

## **2.1** **INTERNET OF THINGS**

The Internet of Things is a novel paradigm shift in IT arena. The phrase “Internet of Things” which is also shortly well-known as IOT is coined from the two words i.e. the first word is “Internet” and the second word is “Things”. The Internet is a global system of interconnected computer networks that use the standard Internet protocol suite (TCP/IP) to serve billions of users worldwide. It is a network of networks that consists of millions of private, public, academic, business, and government networks, of local to global scope, that are linked by a broad array of electronic, wireless and optical networking technologies.

While coming to the Things that can be any object or person which can be distinguishable by the real world. Everyday objects include not only electronic devices we encounter and use daily and technologically advanced products such as equipment and gadgets, but “things” that we do not do normally think of as electronic at all such as food, clothing and furniture. That means here things can be both living things like person or animals.

For starters, (Patel, 2016) the Internet of Things refers to the development of Internet connectivity for everyday devices, which allows them to send and receive data. These devices include everything from your toaster to your car, printer, alarm clock, thermometer, phone, and a variety of machines, but the list doesn’t end there. IoT seeks to connect anything with an on and off switch to the Internet so it can better serve the user.

The Internet of Things essentially works by connecting the devices that surround us to the Internet, or one another, which then allows them to send and receive data to better meet the needs of the user. For example, imagine that when you woke up your alarm lets you know there is traffic, reports this to your car, which then navigates a less congested way to work and receives data on which parking spaces are available, all without your input. Or your printer at work gets an alert that the color cartridge is low and orders more itself. These are just a few examples of how IoT can work, but the potential this technology offers is only in the beginning stages of realization.

The Internet of Things is much more than just a lingering buzzword. It’s the transformation of how all companies do business through the use of connected devices. While it still faces numerous challenges, security being the chief concern, the future of IoT looks bright and businesses investing in and implementing this technology now stand to profit the most.

## **2.2 DEVICE OF INTERNET OF THINGS**

Nowadays, we can see a lot of project that implied IoT concept. There many devices that the developer used to build an IoT project. For this section will focus on the famous IoT platform that popular among IoT developer which is Arduino Uno.

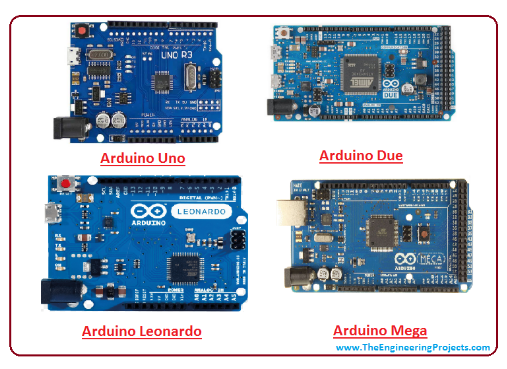
### **2.2.1 Arduino Uno**

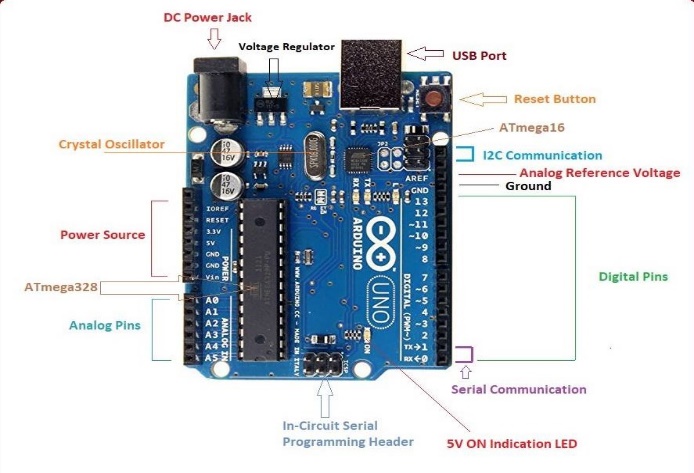
Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code into the board that you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

Experts always strive to introduce innovation in automation that requires minimum effort and gives maximum output. The microcontroller was introduced in the electronics industry with the purpose of making our tasks easy that come with even a remote connection with automation in any way. Microcontrollers are widely used in embedded systems and make devices work according to our needs and requirements. We have already discussed the controllers like 8051, Atmega16, Atmega328 and PIC16F877. Arduino Uno is a very valuable addition in the electronics that consists of USB interface, 14 digital I/O pins, 6 analog pins, and Atmega328 microcontroller. It also supports serial communication using Tx and Rx pins. You should also have a look at this UNO for beginners.

There are many versions of Arduino boards introduced in the market like Arduino Uno, Arduino Due, Arduino Leonardo, Arduino Mega, however, most common versions are Arduino Uno and Arduino Mega. If you are planning to create a project relating to digital electronics, embedded system, robotics, or IoT, then using Arduino Uno would be the best, easy and most economical option.





#### **Figure 2.1 Component of Arduino Uno.**

***Table 2.2 Component and Description of Arduino Uno***

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
|  | USB connecter | This is a printer USB port used to load a program from the Arduino IDE onto the Arduino board. The board can also be powered through this port |
|  | Port Power | The Arduino board can be powered through an AC-to-DC adapter or a battery. The power source can be connected by plugging in a 2.1mm center-positive plug into the power jack of the board. |
|  | Atmega328P Microcontroller | It is the most prominent black rectangular chip with 28 pins. The microcontroller used on the UNO board is Atmega328P by Atmel (a major microcontroller manufacturer). |
|  | Analog | The Arduino UNO board has 6 analog input pins, labeled “Analog 0 to 5.” These pins can read the signal from an analog sensor like a temperature sensor and convert it into a digital value so that the system understands. |
|  | Digital | These pins can be used as either input or output pins. When used as output, these pins act as a power supply source for the components connected to them. When used as input pins, they read the signals from the component connected to them. |
|  | Reset Switch | When this switch is clicked, it sends a logical pulse to the reset pin of the Microcontroller, and now runs the program again from the start. |

**2.2.2 Node MCU ESP8266 WIFI**

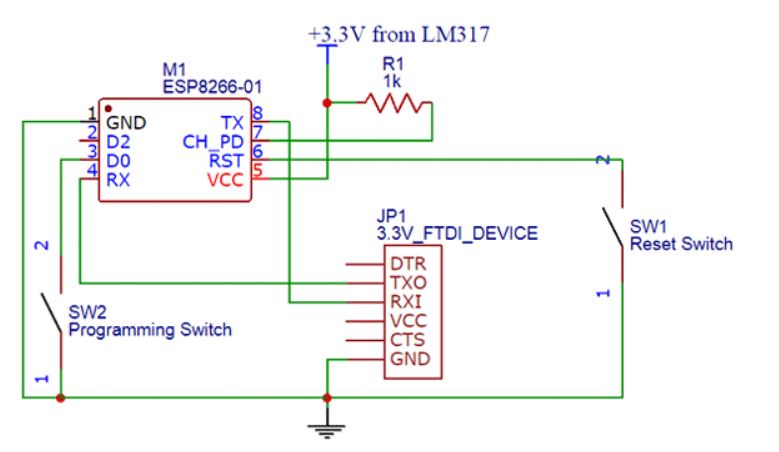
ESP8266 is a highly integrated chip designed for the needs of a new connected world. It offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor.

****

***Figure 2.2 Node MCU ESP8266 WIFI***

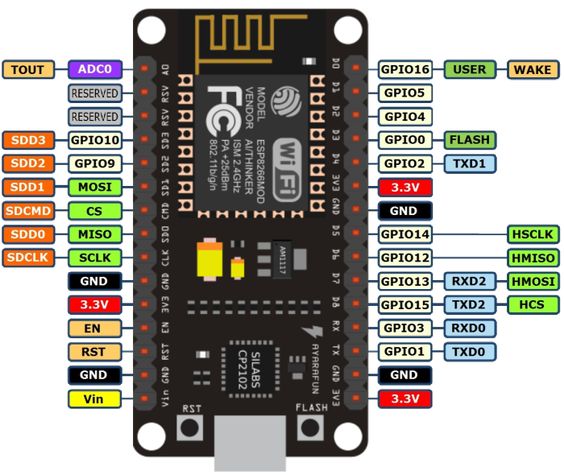
ESP8266 has powerful on-board processing and storage capabilities that allow it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area. The **ESP8266**is a very user friendly and low cost device to provide internet connectivity to your projects. The module can work both as a Access point (can create hotspot) and as a station (can connect to Wi-Fi), hence it can easily fetch data and upload it to the internet making **Internet of Things** as easy as possible. It can also fetch data from internet using API’s hence your project could access any information that is available in the internet, thus making it smarter. Another exciting feature of this module is that it can be programmed using the Arduino IDE which makes it a lot more user friendly. However, this version of the module has only 2 GPIO pins (you can hack it to use up to 4) so you have to use it along with another microcontroller like [Arduino](https://components101.com/microcontrollers/arduino-uno), else you can look onto the more standalone **ESP-12**or **ESP-32** versions. So if you are looking for a **module to get started with IOT** or to provide internet connectivity to your project then this module is the right choice for you.

The **ESP8266 module** works with 3.3V only, anything more than 3.7V would kill the module hence be cautions with your circuits. The best way to program an **ESP-01** is by using the FTDI board that supports 3.3V programming. If you don’t have one it is recommended to buy one or for time being you can also use an Arduino board. One commonly problem that everyone faces with ESP-01 is the powering up problem. The module is a bit power hungry while programming and hence you can power it with a 3.3V pin on Arduino or just use a potential divider. So it is important to make a small voltage regulator for 3.31v that could supply a minimum of 500mA. One recommended regulator is the [LM317](https://components101.com/lm317-pinout-equivalent-datasheet) which could handle the job easily. A **simplified circuit diagram for using the ESP8266-01 module** is given below.



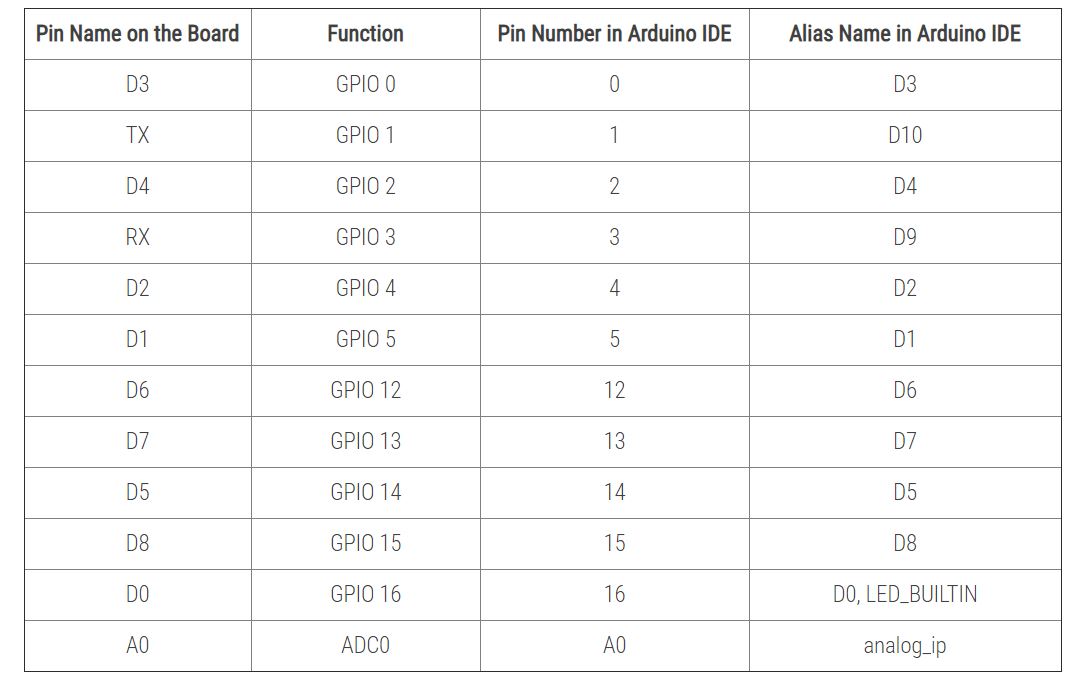
***Figure 2.3 Node MCU graphical***

In the ESP8266 firmware for the Arduino IDE pin numbers are defined as follows:



***Figure 2.3 Node MCU ESP8266 GPIO***

***Table 2.4 GPIO pin in ESP8266***



**Pin Functions**

Pin numbers in the Arduino IDE correspond directly to the ESP8266 GPIO pin numbers. pinMode, digitalRead, and digitalWrite functions work as usual, so to read GPIO2, call digitalRead(2) or its alias name digitalRead(D10). At startup, pins are configured as INPUT. Digital pins 0—15 can be INPUT, OUTPUT, or INPUT\_PULLUP.

Pin 16 can be INPUT, OUTPUT or INPUT\_PULLDOWN\_16 and is connected to the build-in LED. It can be addressed with digitalRead(D0), digitalRead(16) or digitalRead(LED\_BUILDIN).

Pins may also serve other functions, like Serial, I2C, SPI. These functions are normally activated by the corresponding library. The diagram above shows the pin mapping for the popular ESP8266 Node MCU module.

Pin interrupts are supported through attach Interrupt, functions. Interrupts may be attached to any GPIO pin, except GPIO16. Standard Arduino interrupt types are supported: CHANGE, RISING, FALLING.

**Reserved Pins**

GPIO pins 6—11 are not shown on this diagram because they are used to connect flash memory chip on most modules. Trying to use these pins as IOs will likely cause the program to crash.

Note that some boards and modules (ESP-12ED, NodeMCU 1.0) also break out pins 9 and 11. These may be used as IO if flash chip works in DIO mode (as opposed to QIO, which is the default one).

**Vin, 3V3, GND**

Vin is the NodeMcu's voltage input that is connected to its internal voltage regulator allowing an input voltage range of 4.75V to10V. It will be regulated to 3.3V. Alternatively an external voltage source of 3.3V can be directly connected to the NodeMcu's 3V3 pins. The 3V3 pin can be also a voltage source to other components such as LEDs. GND is the common ground of the board.

**Analog Input**

ESP8266 has a single ADC channel available to users. It may be used either to read voltage at ADC pin, or to read module supply voltage (VCC).

To read external voltage applied to ADC pin, use analogRead(A0). Input voltage range is 0 — 1.0V. To read VCC voltage, use ESP.getVcc() while the ADC pin must be kept unconnected. Additionally, the following line has to be added to the sketch:

ADC\_MODE(ADC\_VCC);

This line has to appear outside of any functions, for instance right after the #include lines of your sketch.

**Analog Output**

analogWrite(pin,value) enables software PWM on the given pin. PWM may be used on pins 0 to 16. Call analogWrite(pin,0) to disable PWM on the pin. value may be in range from 0 to PWMRANGE, which is equal to 1023 by default. A value of 0, 512 and 1023 sets the PWM duty cycle to 0%, 50% and 100%, respectively. Optionally, the PWM range may be changed by calling analogWriteRange(new\_range).

PWM frequency is 1kHz by default. Call analogWriteFreq(new\_frequency) to change the frequency. The unit representation is in [Hz]. Example:

analogWriteFreq(32500); //sets PWM frequency to 32.5 kHz

### **Programming**

There are different ways to program the ESP8266, but I'll only cover the method using the Arduino IDE. This is really easy for beginners, and it's a very familiar environment if you've used Arduino boards before.

Just keep in mind that it's not limited to this option: there's also an official SDK available to program it in real C, this is very useful if you want to optimize your code or do some advanced tricks that aren't supported by the Arduino IDE. Another possibility is to flash it with a [LUA](https://github.com/nodemcu/nodemcu-firmware) interpreter, so you can upload and run LUA scripts. Or maybe you're more familiar with Python? Then you should check out the [Micro Python firmware](http://micropython.org/download#esp8266) to interpret Micro Python scripts. I'm sure there's other languages available as well, so just do a quick Google search and write your code in the language of your choice.

**2.2.3 Node MCU Motor Shield**

Node MCU is an open-source development board, based on famous ESP8266 chip (ESP12E or ESP12F), which is a 32-bit controller with integrated WIFI transceiver. And it is very low cost. Best of all, this board is Arduino IDE compatible, you can utilize the millions of example codes and libraries on Node MCU. Besides Arduino IDE or Arduino programming language, Node MCU supports many other languages like LUA script, micro Python etc. With these scripting languages, you can develop your prototype within a few script lines.

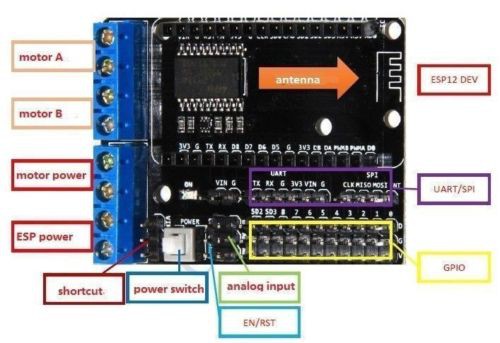
The Node MCU Motor Shield is a driver module for motors that allows you to use to control the working speed and direction of the motor. This Node MCU Motor shield is designed and developed based on ESP-12E from ESP8266, which can be controlled by mobile, PC. This project offers compatible Node MCU Motor shield that can run 2 DC motor and 9-12v battery.

#### **Connections**

1) The Node MCU motor shield digital pins of D0(GPIO 16) and D1(GPIO 5) is connecting to Left Motor.

2) The Node MCU motor shield digital pins of D2(GPIO 4) and D3(GPIO 0) is connecting to Right Motor .

3) The 9-12v photocell battery is connecting to GND and + with Node MCU Motor Shield.



#### **Figure 2.5 Component of Node MCU Motor Shield.**

This shield is designed for the [Node MCU V2 (narrow width)](https://my.cytron.io/p-nodemcu-v2-doit), it comes with L293D motor driver and breakout the GPIO to headers for easy access when connecting to various sensor modules. With the L293D, this shield supports the control of two (2) brush motor, start, stop, brake, direction and also speed. The shield also extends the GPIO of Node MCU to header pins in SVG sequence (RC servo alike), not to forget the UART, SPI, Power and GND pins. There are options to power the board separately or with single power source. The board also comes with a power LED indicator and latching push button as power switch. This shield is very convenient and cheap to build or DIY a WIFI RC smart car or any type of robotics or even UAV platform.

This shield board is driven by the special excent large power full-bridge chip **L293DD** from the famous Stmicro electronics company, which can directly drive 2-channels DC motors or one-channel stepper motor. The driven current can be arrived at 1.2A. This board is generated with national layout, SMT ensuption, and convenient installation.

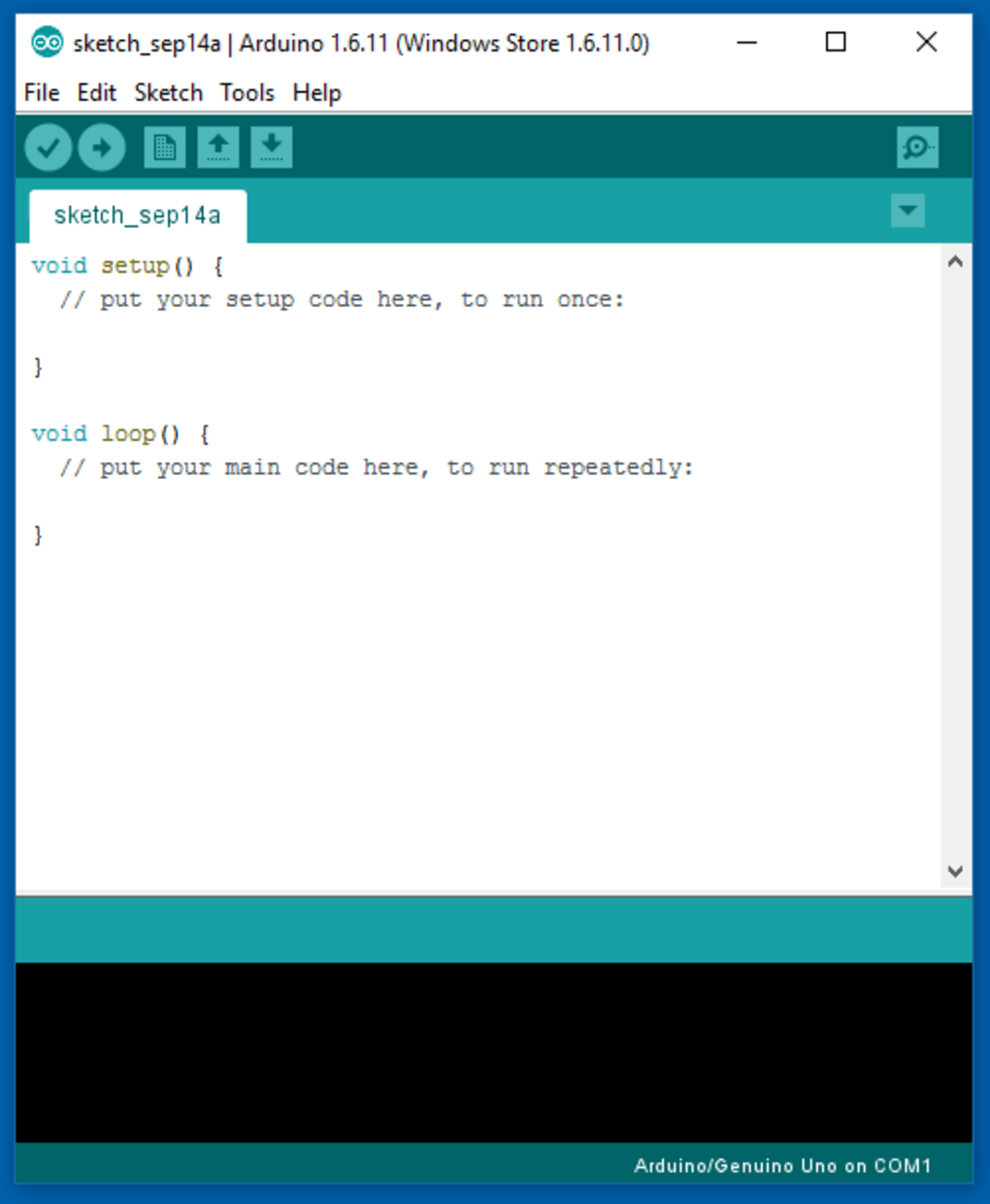
In this motor shield board, the IO port of ESP-12E Dev Kit is used as the control port. The logic chip configured inside can finish IC driven. Thus, the shield board has four ports: D1, D2, D3, and D3, which are used as PWMA (motor A), PWMB (motor B), DA (direction of motor A), and DB (direction of motor B), respectively.

In addition, this shield board has many pins, such as VIN, 3.3V, DIO, AIO, SDIO, UART, SPI, RST, and EN, thus can conveniently connect all kinds of sensors (e.g., temperature and humidy, buzzer, light, relay sensor, etc.). The board is developed by the humanized design with a power switch, and thus user can control the on-off of power conveniently.

**2.2.4 Arduino IDE**

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures.

User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. In October 2019 the Arduino organization began providing early access to a new Arduino Pro IDE with debuggingand other advanced features



***Figure 2.6* Arduino IDE**

**2.4.5 Cable USB 2.0**

Universal Serial Bus, [USB](https://en.wikipedia.org/wiki/USB): connectors, cabling, and power. The initial versions of the USB standard specified connectors that were easy to use and that would have acceptable life spans; revisions of the standard added smaller connectors useful for compact portable devices. Higher-speed development of the USB standard gave rise to another family of connectors to permit additional data paths.

All versions of USB specify cable properties; version 3.X cables include additional data paths. The USB standard included power supply to peripheral devices; modern versions of the standard extend the power delivery limits for battery charging and devices requiring up to 100 watts. USB has been selected as the standard charging format for many mobile phones, reducing the proliferation of proprietary chargers.

****

***Figure 2.7* Cable USB**

**2.4.6 Cable Jumper**

A jump wire (also known as jumper wire, or jumper) is an [electrical wire](https://en.wikipedia.org/wiki/Electrical_wire), or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a [breadboard](https://en.wikipedia.org/wiki/Breadboard) or other prototype or test circuit, internally or with other equipment or components, without soldering.Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the [header connector](https://en.wikipedia.org/wiki/Pin_header#Header_connector) of a circuit board, or a piece of test equipment.

****

***Figure 2.8* Cable Jumper (Male-Male, Female-Female, Male-Female)**

**2.4.7 Ultrasonic Sensor**

Ultrasonic sensors work by emitting sound waves at a frequency too high for humans to hear. They then wait for the sound to be reflected back, calculating distance based on the time required. This is similar to how radar measures the time it takes a radio wave to return after hitting an object. While some sensors use a separate sound emitter and receiver, it’s also possible to combine these into one package device, having an ultrasonic element alternate between emitting and receiving signals. This type of sensor can be manufactured in a smaller package than with separate elements, which is convenient for applications where size is at a premium.



***Figure 2.9 Ultrasonic Sensor***

While radar and ultrasonic sensors can be used for some of the same purposes, sound-based sensors are readily available they can be had for just a couple dollars in some cases and in certain situations, they may detect objects more effectively than radar. For instance, while radar, or even light-based sensors, have a difficult time correctly processing clear plastic, ultrasonic sensors have no problem with this. In fact, they’re unaffected by the color of the material they are sensing.

On the other hand, if an object is made out of a material that absorbs sound or is shaped in such a way that it reflects the sound waves away from the receiver, readings will be unreliable. If you need to measure the specific distance from your sensor, this can be calculated based on this formula:

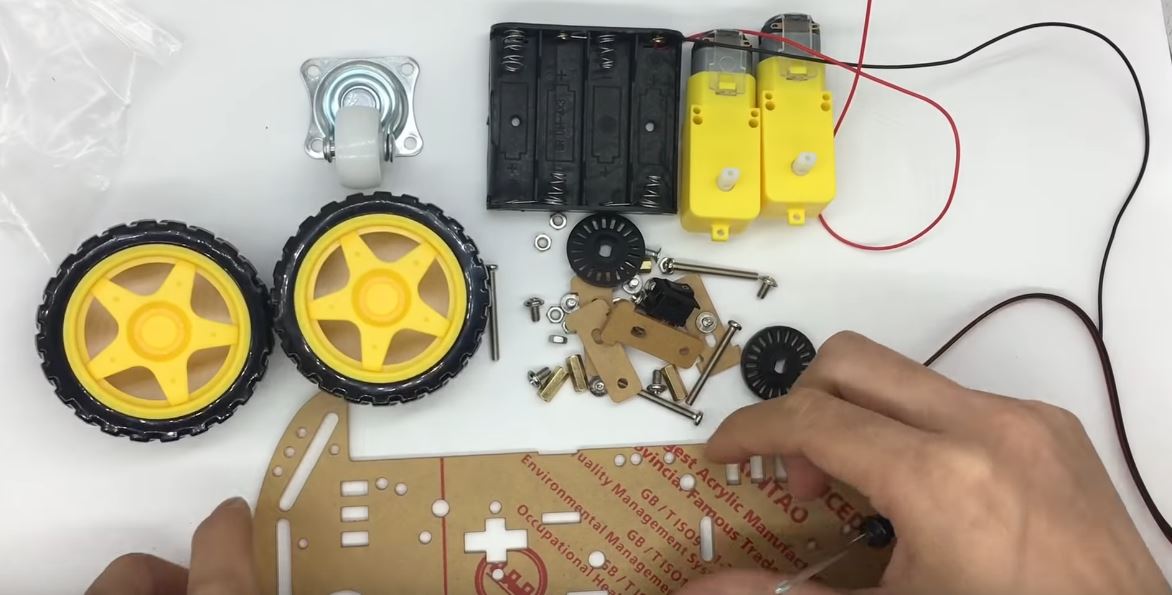
**Distance = ½ T x C**

(**T** = **Time** and **C** = **the speed of sound**)

At 20°C (68°F), the speed of sound is 343 meters/second (1125 feet/second), but this varies depending on temperature and humidity. Specially adapted ultrasonic sensors can also be used underwater. The speed of sound, however, is 4.3 times as fast in water as in air, so this calculation must be adjusted significantly.

**2.4.8 2WD Smart Robot Car Chassis**

2WD Robot Car Chassis comes with one screwdriver, two DC motors, two wheels and one acrylic car chassis. There is also a small plastic bag that comes with a battery holder, a small wheel for the front, some bolts and screws, four wires and other required components to assemble the robot.



***Figure 2.7 2WD Chassis Car Kit***

### **2.4.9 LED**

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, sensing surroundings.

***Figure 2.7.3* LED**

***Figure 2.7.1 LED***

Although this kit comes with the needed wires, since they are a bit short and not very flexible, we’ve decided to replace them with other wires, because we can easily adjust their length. You’ll need one red wire and one black wire for each motor. You can use a wire cutter to cut the wires to your desired length. After preparing all four wires, DC motors should be to solder them. Tin the DC motor pins.

### Motor:

***Figure 2.8 Table of Motor Chassis***

|  |  |  |  |
| --- | --- | --- | --- |
| Voltage | DC 3V | DC 5V | DC 6V |
| Current | 100 MA | 100MA | 120MA |
| Reduction rate | 48:1 | | |
| RPM (With tire) | 100 | 190 | 240 |
| Tire Diameter | 66mm | | |
| Car Speed(M/minute) | 20 | 39 | 48 |
| Motor Weight (g) | 50 | | |
| Motor Size | 70mm\*22mm\*18mm | | |
| Noise | <65dB | | |

Note: Motor power supply is 3V~6V. All the parameters above are tested without load.

# **CHAPTER 3**

## **PROJECT METHODOLOGY**

**3.1 Reason of Choosing Project Methodology**

This methodology used for this project is adapted from the Agile methodology. For IoT system development, this methodology is the most suitable because of the couple of reasons. First reason is that for the time when this project made, the IoT system development were still in the beginning phase where there were still fresh in the market and not a popular choice for public usage especially for the one who lived in developing country. Because of that, the improvements are still needed so that the system will give the maximum satisfaction to the user or client. These are the features that contained from agile methodology. The reviews and evaluations of the customer are the crucial part on how the agile works because the development team need to know time to time on how to develop this type of technology that still new and still target to give a full satisfaction for the customer. Other reason for choosing this methodology is the development of IoT need a flexibility of the change and evolution of the product thus increase development productivity and the product is up to date.

**3.2 Overview Project Methodology**

Since the chosen methodology is Agile, there are certain thing that need to be understand on what is agile. Basically, agile methodology is a practice that approach continuous iteration of development throughout the software development lifecycle of the project. Development activities is concurrent unlike the Waterfall model that practice sequent way from the beginning to the end. Beside of the repetition process, this methodology basically consists of 4 phases which is requirement analysis, design, develop and testing. Each phase has their own activities, deliverables and objective that should be met. The part will be explained on what are the detail of each phases.

This project are using Agile methodology. The [Agile methodology](https://agilemanifesto.org/) was created as a direct response from critics of Waterfall who felt that there was too much room for problems to remain unnoticed until the project neared completion. With Agile development, teams break each feature into the smallest discernible chunk of work and produce incremental value over time. Agile focuses on delivering working software with each release.

However, in the IoT world, consumer expectations are fueling the need for more communication between project teams. Hardware and software design and development need to happen simultaneously with a great deal of communication between team members with different skills. The Agile methodology focuses on collaboration between teams, which results in getting products to market faster and making more product updates and releases.

As an internal process, Waterfall focuses very little on the end user of a product. Its main purpose is to help internal teams move more efficiently through the phases of a project. However, it’s often important to involve clients in a project, adding opinions and clarifying what they want as the project moves forward. In these cases, consider an agile methodology to better anticipate change and keep stakeholders informed through the life of the project. By involving stakeholders throughout the process, you lower the risk of late requests for change throwing off your project deadlines.

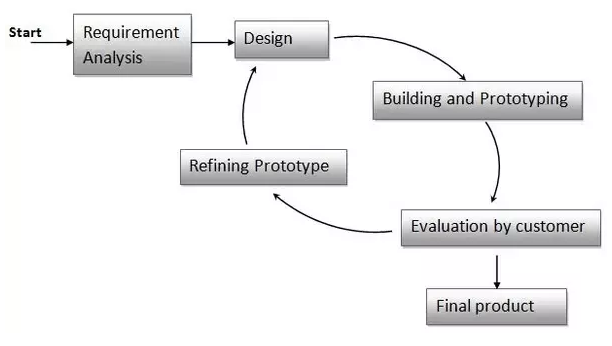
Agile encourages or requires frequent communication between developers and those who will ultimately accept and use the product. This pays major dividends when effective. For example, feedback can be incorporated into future iterations as increments are delivered and reviewed by users, a Product Owner, or both. False assumptions made by developers can be recognized early, reducing impact. Agile gives us continual opportunities to learn via this feedback.

In contrast, Agile methodologies and continuous delivery are particularly well suited for dealing with the demands of the connected device. With Agile, testing becomes an essential component of each phase of the development process, with quality being “baked in” at every stage. This is especially crucial for IoT development projects because real-world conditions are unpredictable. Continuous testing saves time, money, and frustration.

In comparison, Agile is built to account for changing requirements, making it easy to adjust your approach to target high-priority issues. Working iteratively gives you more flexibility for change and evolution during product lifecycles. Sprint projects increase manufacturer productivity by pinpointing low-hanging, solvable problems and directing teams to focus on building the most essential features.

In most cases, the steps of the prototyping model are as follows, firstly, the new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the departments or aspects of the existing system. Secondly, a preliminary, simple design is created for the new system. Thirdly, first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product. Fourthly, the users thoroughly evaluate the first prototype and note its strengths and weaknesses, what needs to be added and what should to be removed. The developer collects and analyzes the remarks from the users.

Fifth, the first prototype is modified, based on the comments supplied by the users, and a second prototype of the new system is constructed. Sixth, the second prototype is evaluated in the same manner as was the first prototype. Seventh, the preceding steps are iterated as many times as necessary, until the users are satisfied that the prototype represents the final product desired. Eighth, the final system is constructed, based on the final prototype. Ninth, the final system is thoroughly evaluated and tested. Routine maintenance is carried out on a continuing basis to prevent large-scale failures and to minimize downtime.



***Figure 2.8.1* Prototype Model**

## **3.3 Requirements Gathering and Analysis Phase**

The **prototyping model** is applied when detailed information related to input and output requirements of the system is not available. In this model, it is assumed that all the requirements may not be known at the start of the development of the system. It is usually used when a system does not exist or in case of a large and complex system where there is no manual process to determine the requirements.

This model allows the users to interact and experiment with a working model of the system known as **prototype.**The prototype gives the user an actual feel of the system. At any stage, if the user is not satisfied with the prototype, it can be discarded and an entirely new system can be developed. A prototyping model starts with requirement analysis. In this phase, the requirements of the system are defined in detail. During the process, the users of the system are interviewed to know what is their expectation from the system.

**3.4 Quick Design Phase**

A quick design is carried out and a prototype is built. The developed prototype is submitted to the customer for evaluation. Based on the customer feedback, the requirements are refined and the prototype is suitably modified. The second phase is a preliminary design or a quick design. In this stage, a simple design of the system is created. However, it is not a complete design. It gives a brief idea of the system to the user. The quick design helps in developing the prototype.

### **3.4.1 System Requirement**

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to meet these requirements can result in installation problems or performance problems. The former may prevent a device or application from getting installed, whereas the latter may cause a product to malfunction or perform below expectation or even to hang or crash.

### **3.4.1.1 Hardware Requirement**

A list of hardware requirements for developing proposal application is shown in table below.

***Table 3.1* Hardware Requirement List.**

|  |  |  |
| --- | --- | --- |
| **Hardware** | **Specification** | **Description** |
| Processor | Intel® Celeron® N3550  CPU @ 1.10 GHz | The CPU performs basic arithmetic, logic, controlling, and input/output operations specified by the instructions. |
| RAM | 4.00 GB | Stores information on the computer including files and software programs. |
| Operating System | Microsoft Windows 7 and Microsoft Windows 10  (64-bit) | The operating system is the requirement of many online programs. |
| Input Device | Keyboard  Mouse | The keyboard allows the user to type letters and numbers and the mouse allows the user to position the cursor. |
| ESP8266 | NodeMCU ESP8266 | The ESP8266 is an open-source lowcost WIFI microchip with TCP/IP stack and microcontroller developed by Arduino.cc.  This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using [Hayes](https://en.wikipedia.org/wiki/Hayes_command_set)-style commands. |
| Node MCU Motor Shield | Motor Shield | The Node MCU Motor Shield is a driver module for motors that allows you to use to control the working speed and direction of the motor.This Node MCU Motor shield is designed and developed based on ESP-12E from ESP8266, which can be controlled by mobile |

### **3.4.1.2 Software Requirement**

A list of software requirements for developing proposal application is shown in table below.

***Table 3.2* Software Requirement List.**

|  |  |  |
| --- | --- | --- |
| **Software** | **Specification** | **Description** |
| Cross-platform Application | Arduino IDE | The Arduino Integrated Development Environment is a cross-platform application that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards |
| Application Platform | Blynk | Blynk is a Platform with IOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. It is a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets. |
|  | Thinger.io | Thinger.io platform is an Open Source platform for the Internet of Things, it provides a ready to use scalable cloud infrastructure for connecting things. |

**3.5 Build A Prototype Phase**

By using this prototype, the client can get an “actual feel” of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system. Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements. In this phase, an actual prototype is designed based on the information gathered from quick design. It is a small working model of the required system.

**3.6 Initial User Evaluation Phase**

Testing a prototype / developed design is a very important part of the design and manufacturing process. Testing and evaluation, simply confirms that the product will work as it is supposed to, or if it needs refinement. In general, testing a prototype allows the designer and client to assess the viability of a design.

In this stage, the proposed system is presented to the client for an initial evaluation. It helps to find out the strength and weakness of the working model. Comment and suggestion are collected from the customer and provided to the developer.

**3.7 Refining Prototype Phase**

Prototyping is a development methodology in which a model is quickly constructed to test or illustrate design features and ideas, in order to gather user feedback. Subsequent models are created by refining earlier versions, with the aim of convergence on the desired end product.

If the user is not happy with the current prototype, you need to refine the prototype according to the user's feedback and suggestions.

This phase will not over until all the requirements specified by the user are met. Once the user is satisfied with the developed prototype, a final system is developed based on the approved final prototype.

**3.8 Implement Product and Maintain Phase**

The implement product and maintenance phase involves making changes to hardware, software, and documentation to support its operational effectiveness. It includes making changes to improve a system's performance, correct problems, enhance security, or address user requirements.

Once the final system is developed based on the final prototype, it is thoroughly tested and deployed to production. The system undergoes routine maintenance for minimizing downtime and prevent large-scale failures.

**3.9 Conclusion**

In conclusion, the prototype model is used because of its advantages. It gives the clear picture of software what is being developed and not to the client earlier and before the implementation stage as well. It helps in saving the later part of issues that might occur. It helps in communication of client and developer to make the product as per requirement and easier to understand them as well for the developers from the viewpoint of the customer. The prototype model provides different types of model as well.

***Figure 2.9 methodology of Smartphone Control Car***

USER SIDE

PIC MICROCONTROLLER

COMMUNICATION LINK

ESP8266WIFI

BLYNK APPLICATION

ANDROID PHONE

SERVO MOTOR

STEERING

# **CHAPTER 4**

**DEVELOPMENT**

## **4.1 Introduction**

This chapter will discuss about the overview of proposed design and the development of the Smartphone Control Car project. The design and development phase are the next phase after the requirement phase. The process of the development can be done by using the available software.

In addition, this chapter also discuss about the tools and technology used in the development of the project. Moreover, this chapter will discuss about the interface of the system would flow so that the reader will easily understand.

### **4.1.1 Problems**

Problem that I had been through:

1. Error when uploading sketch
2. Port not found
3. Error in coding
4. Troubleshoot problem

**4.2 Requirement Gathering**

Requirement gathering is the initial phase of the Prototype Methodology where the information will be gathered required to the project title. Requirements gathering and analysis, a prototyping model starts with requirement analysis. In this phase, the requirements of the system are defined in detail. During the process, the users of the system are interviewed to know what is their expectation from the system. This will make the project clearly understand by the developer. The second phase is design.

**4.3 Design**

The second phase is a preliminary design or a quick design. In this stage, a simple design of the system is created. However, it is not a complete design. It gives a brief idea of the system to the user. The quick design helps in developing the prototype.

Next, the third phase is build a prototype.

**4.4 Build a Prototype**

In this phase, an actual prototype is designed based on the information gathered from quick design. It is a small working model of the required system.

**4.5 Development**

Development is the crucial phase as the developer will start build the system on this phase. During this phase, there are 2 major software that been used to develop Smartphone Control Car, Arduino IDE for microcontroller and Blynk application for the notifications and also Thinger.io for the database.

**4.5.1 System Component**

Microcontroller contains the functionality of the system. The main function of the Smartphone Control Car is the control function where the user can just control the car using smartphone at anyplace, anytime, anywhere, because the fan and bulb can automatically switch on and off based on the WIFI connection and will be control the car through Blynk app productively

***Figure 2.10 WIFI car code***

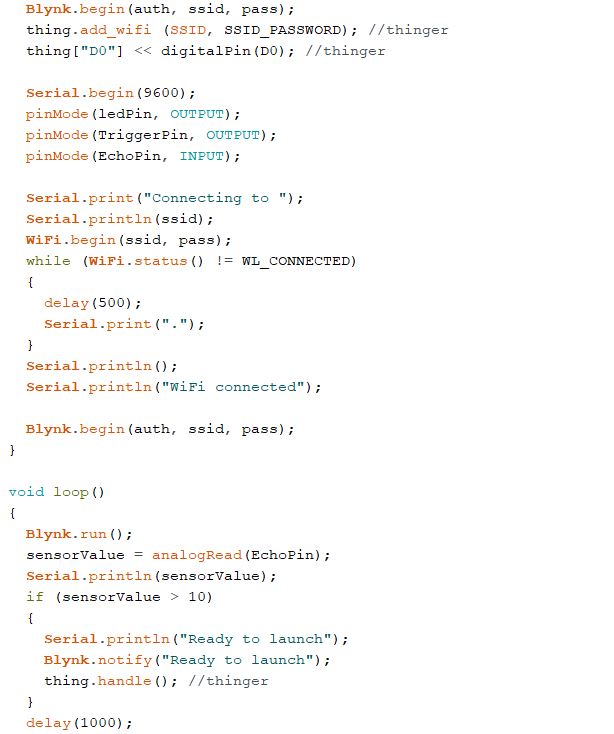
******

******

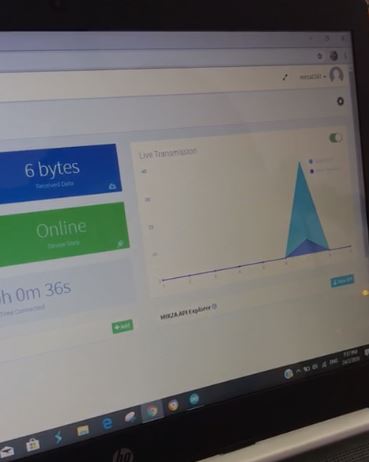


**Figure 2.11 Notifications Sensor code**

****

****

***Figure 4.5 Database for Notifications Sensor***

******

**4.6 Components**

In this section will show how the components of the project been connected and the what type of the components that been used to produce this project. Various type of components used to complete the Smartphone Car Control project based on table 4.6.1

***Table 4.6.1.1 The Components used in Smartphone Car Control***

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Description** |
| 1 | Node MCU ESP8266 WIFI | The ESP8266 is a low-cost Wi-Fi microchip, with a full TCP/IP stack and microcontroller  capability. |
| 2 | Node MCU Motor Shield | The Node MCU Motor Shield is a driver module for motors that allows you to use to control the working speed and direction of the motor. This Node MCU Motor shield is designed and developed based on ESP-12E from ESP8266, which can be controlled by mobile |
| 3 | 2WD Chassis Car Kit | Stability of the car |
| 4 | Ultrasonic Sensor | Send notifications through sensing. |
| 5 | Battery port | A power source for Node MCU motor shield. |
| 6 | Smartphone | Control the car through BLYNK application\ |

### **4.6.1 Type of Components**

These are a component that been used for build Smartphone Control Car project. Combination of this component will make Smartphone Control Car come to life and can be operated based on the requirements that been gathered.



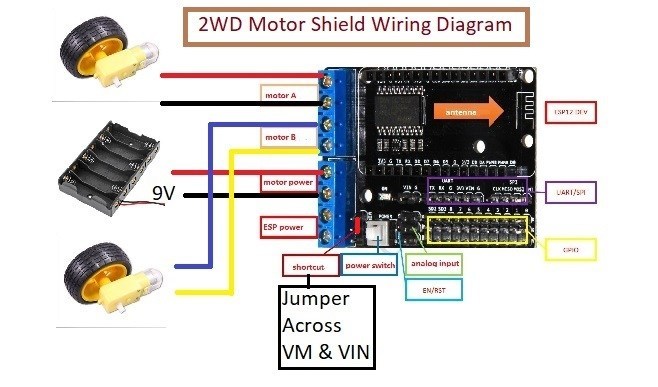
***Figure 4.6* System Prototype**



***Figure 4.6.1*.*1* Sensors Prototype**

### **4.6.2 Schematic of the Components**

Based on the **Figure 4.6.1.2**, the main component that seem connected to most of other components is Node MCU ESP8266 WIFI. The component that connects to the ESP8266 WIFI are Node MCU motor shield, Ultrasonic sensor, LED.



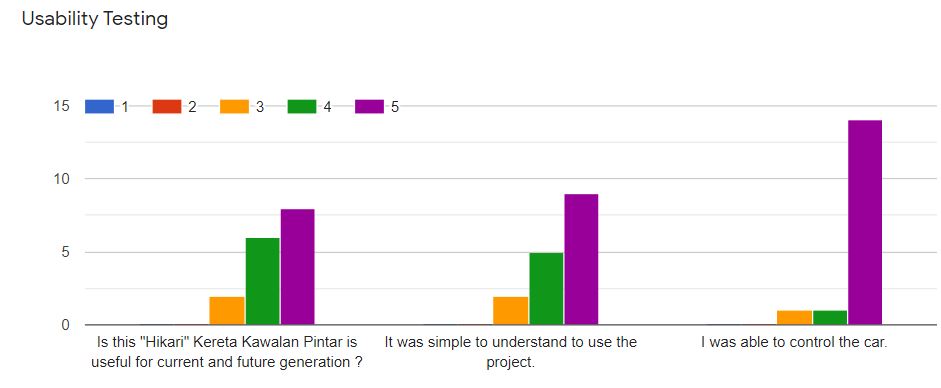
***Figure 4.6.1.2* Schematic Diagram**

## **4.7 Testing and Analysis**

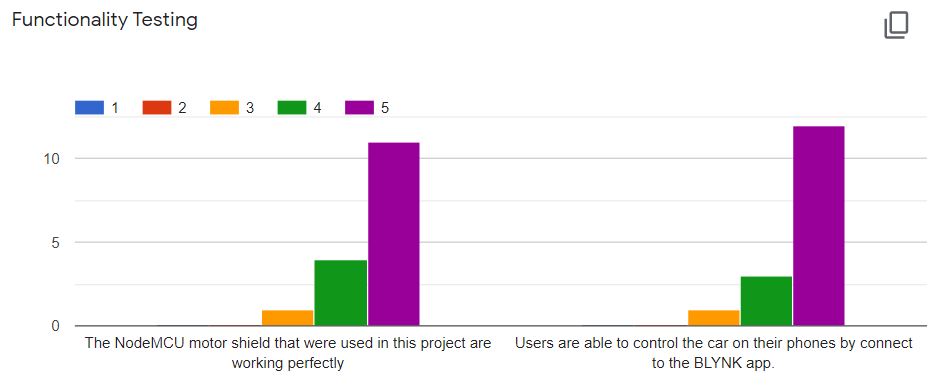
A testing been made to see how the Smartphone Car Control work this include:

1. Motor gear can switch on and off automatically based on the WIFI or mobile connection.
2. Ultrasonic sensor detecting the surroundings and send notifications.
3. LED as indicator.

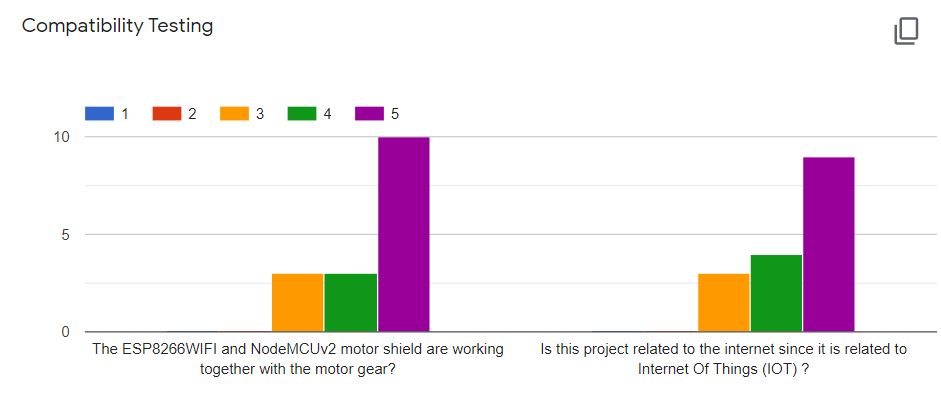
## **Responses**



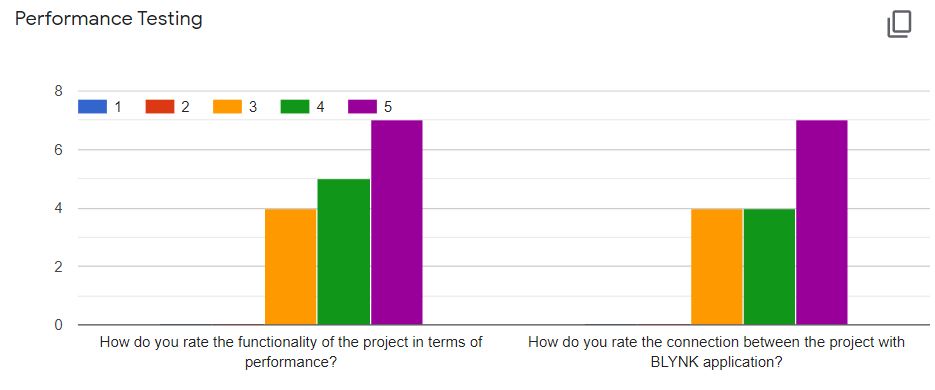
***Figure 4.6.2 Usability Testing Graph***



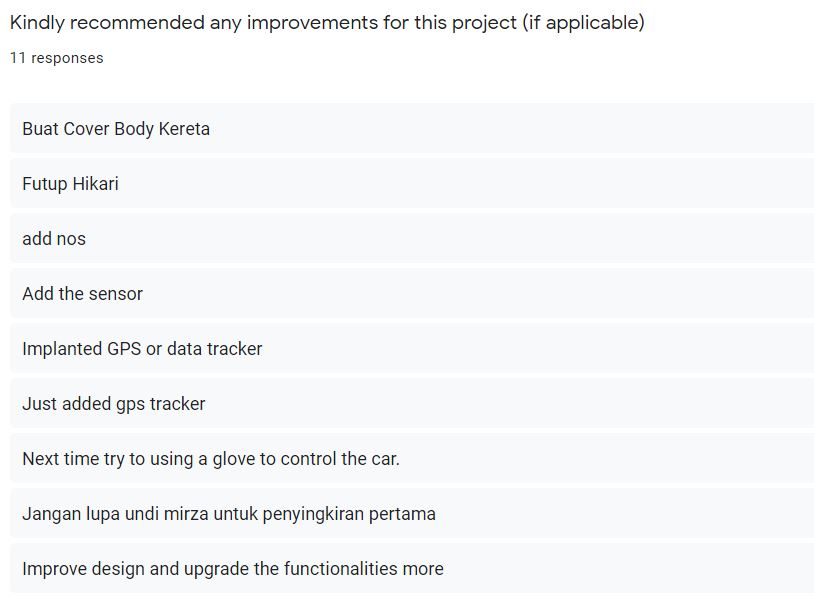
***Figure 4.6.3 Functionality Testing Graph***



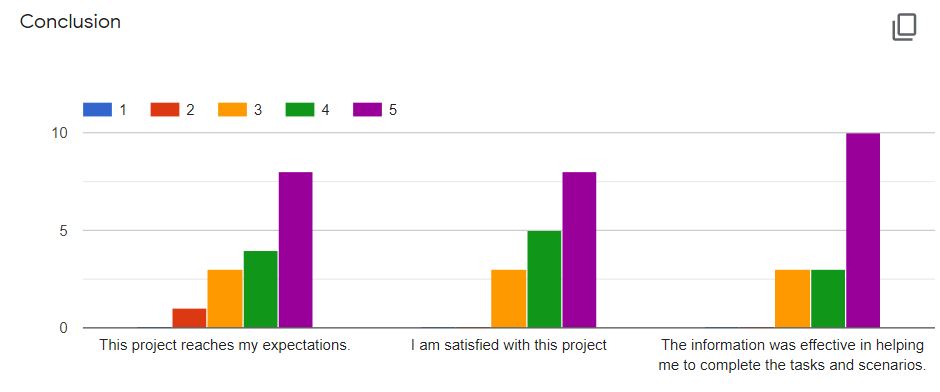
***Figure 4.6.4 Compatibility Testing Graph***



***Figure 4.6.5 Performance Testing Graph***



***Figure 4.6.6 Recommended Answer***



***Figure 4.6.7*** ***Conclusion Graph***

## **CHAPTER 5**

## **Summary** In this paper the component designed is easy to maintain, affordable and portable. Thus having result for days of the testing, a movement of gear motor was moved clearly. The sensor also can sensoring surroundings within a similar range. For recommended for users in future, another recommended is the GPS tracker should be used as location tracking system. It is because of the untimely failure of location finding to enhance the efficiency of the system. In addition, design represents as a ‘face’ of the system. Hence in this chapter, the design of Smartphone Control Car was discussed where which architecture system to apply and what is the flow of the system. Other than that, storyboard or user interface was created to see the function for each page of the system and how the design for the system should be.

**REFERENCES**

Anonymous. (n.d.) 15 Arduino Uno Breadboard Projects. Retrieved from

<https://www.makerspaces.com/15-simple-arduino-uno-breadboard-projects/>

WIFI Car Controlled Robot | NodeMCU ESP8266.

<https://www.youtube.com/watch?v=dx9b9T9Addo/>

Smartphone controlled NodeMCU (ESP8266) 4WD Robot CAR.

<https://www.youtube.com/watch?v=mJEftoeSBOU>

Wifi Robot Car/RC car NodeMCU ESP8266.

<https://www.youtube.com/watch?v=vS-2dj_Knak>

Node MCU Motor Shield L293D for ESP12E + Blynk

<https://www.youtube.com/watch?v=sEjhM3cMlhc>

#046 ENG Blynk & Motor Shield Board L293D for ESP12E

<https://www.youtube.com/watch?v=7chv4fYqpg0>

Motor Driver Shield Board Module L293D for ARDUINO.

<https://www.youtube.com/watch?v=aLGFBG-VDdo>

[Android Developers Guide. Android Architecture0. [Online]2013.

<https://.android.com/about/versions/index.html>.

Anonymous(n.d.) *Build your own Adjustable Electronic DC Load using Arduino*.

from <https://circuitdigest.tumblr.com/>

NodeMcu Motor Shield.

<https://www.youtube.com/watch?v=_fgdHFobCHY>

ARDUINO BLYNK LESSON 2: NOTIFICATION.

<https://www.youtube.com/watch?v=ZrmcPDJQms0>

What is Blynk and how does it work?

<https://www.youtube.com/watch?v=YULg0IoqoZM>

Blynk Arduino Tutorial – Blynk and Arduino Home Automation

<https://www.youtube.com/watch?v=osACD1WX6sg>

Tutorial: Control Arduino over USB with Blynk app. No shield required.

<https://www.youtube.com/watch?v=fgzvoan_3_w>

ESP8266/NodeMCU using HC-SR04 Ultrasonic Sensor and Blynk App

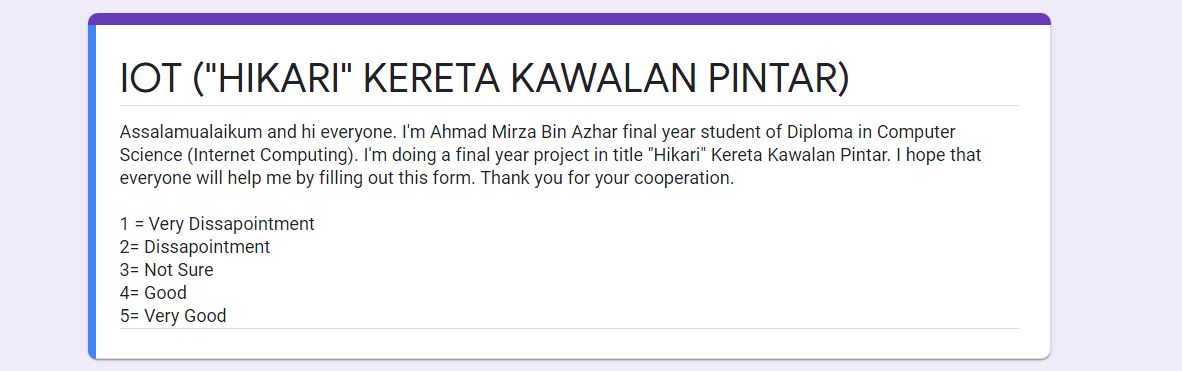
<https://www.youtube.com/watch?v=YqRaeT8s3m0>

ESP8266 nodemcu Arduino + Blynk send notification EVENTOR Widget

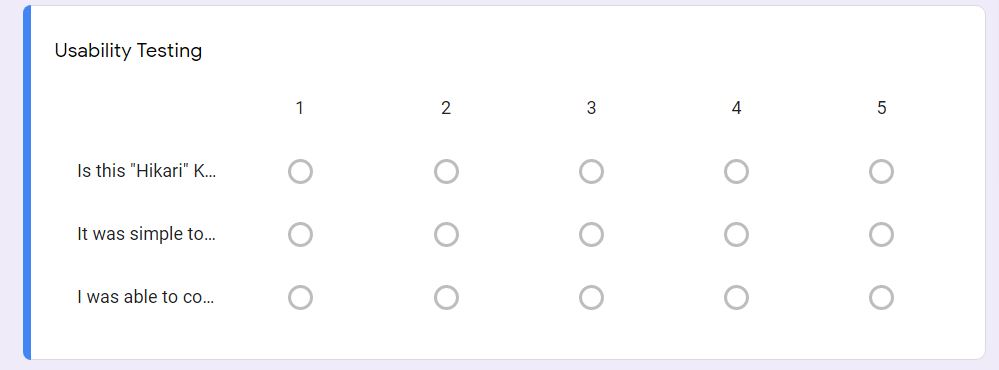
<https://www.youtube.com/watch?v=do32r_Qc5Mw>

**APPENDIX I**

**QUESTION SECTION**

****

***Figure 4.6.8*** ***Introduction of Smartphone Control Car Question***

******

***Figure 4.6.8 Usability Testing Question***

******

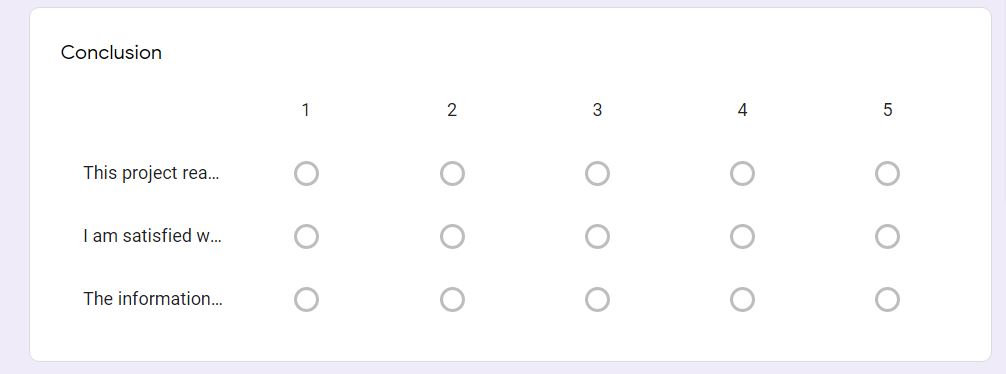
***Figure 4.6.9*** ***Functionality Testing Question***

******

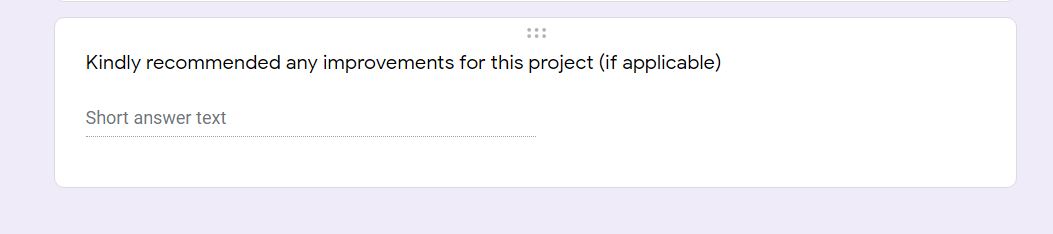
***Figure 4.7.0*** ***Compatibility Testing Question***

******

***Figure 4.7.1 Performance Testing Question***

******

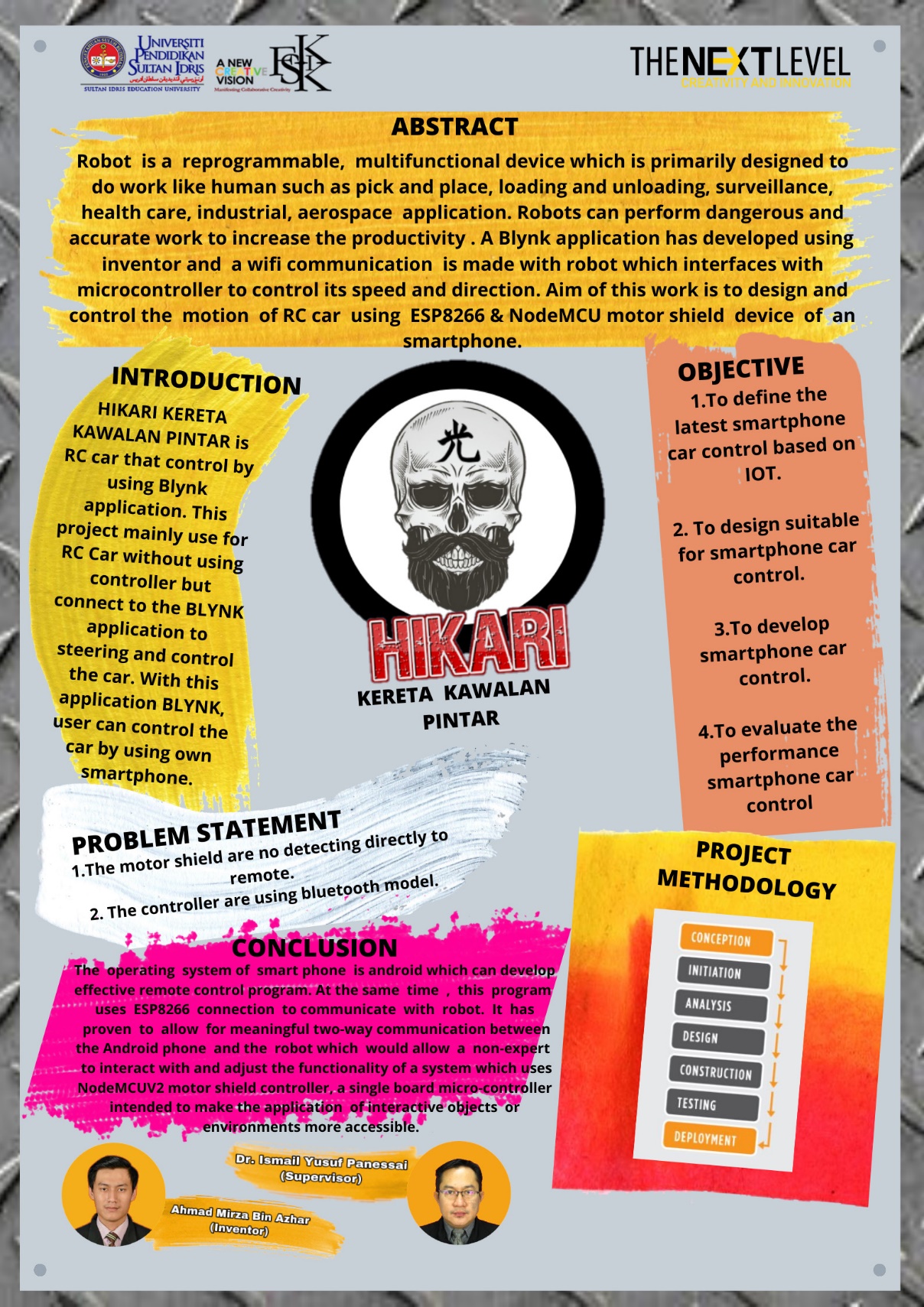
***Figure 4.7.2*** ***Conclusion Question***

******

***Figure 4.7.3*** ***Recommended Improvements Question***

**APPENDIX II**

**POSTER**

****

***Figure 4.7.4 Poster of Project***

**APPENDIX III**

**CODING I**

***Figure 2.10 WIFI car code***

******

******

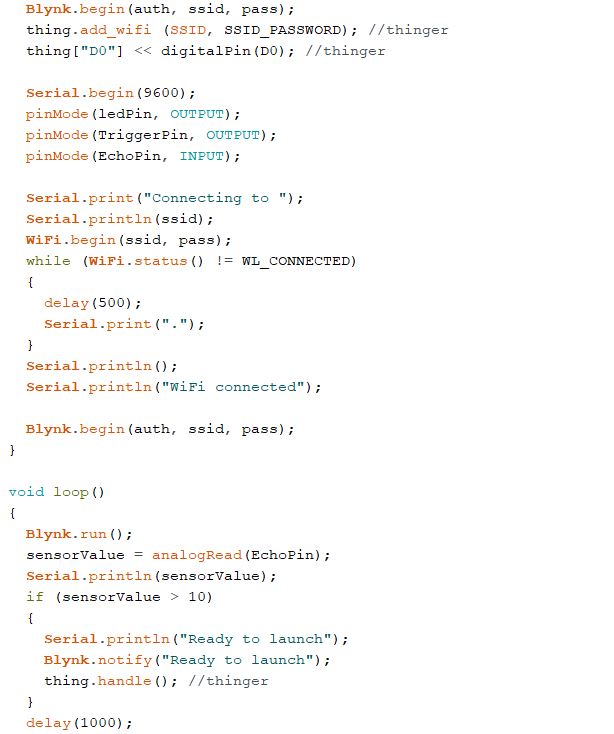


**APPENDIX IV**

**Coding II**

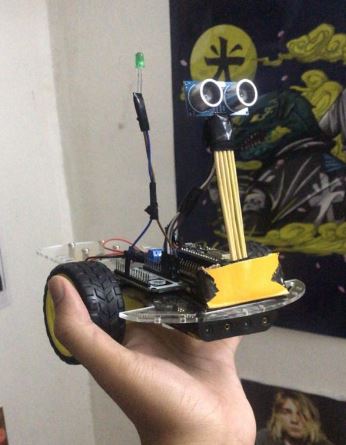
**Figure 2.11 Notifications Sensor code**

****

****

**APPENDIX V**

**PROTOTYPE PROJECT**

****

***Figure 4.7.4 Prototype of Project***

***APPENDIX VI***

***PROGRESS OF PROJECT***

******

***Figure 4.7.5 Smartphone Controller through Blynk apps***

******

***Figure 4.7.6 Start to steering the Car***

******

***Figure 4.7.6 Car is working and ready to steering***

**APPENDIX VII**

**EXHIBITION**

****

***Figure 4.7.7 Smartphone Car Control are display in Exhibition Day***

***Figure 4.7.8 Smartphone Car Control was controlled by***

***visitors and coordinator in Exhibition Day***

******

******

******

******

**TABLE OF CONTENT**

**PAGE**

**ACKNOWLEDGEMENTS I**

**ABSTRACT II**

**ABSTRAK III**

**TABLE OF CONTENT IV**

**LIST OF TABLES VIII**

**LIST OF FIGURES VIII**

**LIST OF APPENDIX XI**

**CHAPTER 1 INTRODUCTION**

* 1. Project Background 1
  2. Problem Statement 2
  3. Research Question 3
  4. Research Aim 4
  5. Research Objective 5
  6. Conclusion 6
  7. Research Significance 7

**CHAPTER 2 LITERATURE VIEW**

2.1 Internet of Things 8

2.2 Device of Internet of Things 10

2.2.1 Arduino Uno

2.2.2 Node MCU ESP8266 WIFI 12

2.2.3 Node MCU Motor Shield 18

2.2.4 Arduino IDE 19

2.4.5 Cable USB 2.0 20

2.4.6 Cable Jumper 21

2.4.7 Ultrasonic Sensor 22

2.4.8 2WD Smart Robot Car Chassis 23

2.4.9 LED 24

**CHAPTER 3 PROJECT METHODOLOGY**

3.1 Reason of Choosing Project Methodology 26

3.2 Overview Project Methodology

3.3 Requirements Gathering and Analysis Phase 29

3.4 Quick Design Phase 30

3.4.1 System Requirement

3.4.1.1 Hardware Requirement 31

3.4.1.2 Software Requirement 33

3.5 Build A Prototype Phase 34

3.6 Initial User Evaluation Phase

3.7 Refining Prototype Phase

3.8 Implement Product and Maintain Phase 35

3.9 Conclusion

**CHAPTER 4 DEVELOPMENT**

4.1 Introduction 38

4.1.1 Problems

4.2 Requirement Gathering 39

4.3 Design

4.4 Build a Prototype

4.5 Development

4.5.1 System Component 40

4.6 Components

4.6.1 Type of Components 46

4.6.2 Schematic of the Components 48

4.7 Testing and Analysis

4.8 Responses 50

**CHAPTER 5 CONCLUSION**

Summary 53

**LIST OF TABLES**

**TABLE PAGE**

Table 2.2 Component and Description of Arduino Uno 16

Table 2.4 GPIO pin in ESP8266 19

Table 2.8 Table of Motor Chassis 32

Table 3.1 Hardware Requirement List. 38

Table 3.2 Software Requirement List 40

Table 4.6.1 The Components used in Smartphone Car Control 52

**LIST OF FIGURES**

**FIGURE PAGE**

Figure 2.1 Component of Arduino Uno 15

Figure 2.2 Node MCU ESP8266 WIFI 17

Figure 2.3 Node MCU graphical 19

Figure 2.4 Node MCU ESP8266 GPIO 19

Figure 2.5 Component of Node MCU Motor Shield 24

Figure 2.6 Arduino IDE 27

Figure 2.7 Cable USB 2.0 28

Figure 2.8 Cable Jumper (Male-Male, Female-Female, Male-Female) 29

Figure 2.9 Ultrasonic Sensor 29

Figure 2.7 2WD Chassis Car Kit 31

Figure 2.7.1 LED 31

Figure 2.8 Table of Motor Chassis 32

Figure 2.8.1 Prototype Model 36

Figure 2.9 Methodology of Smartphone Car Control 43

Figure 2.10 WIFI Car Code 46

Figure 2.11 Notifications Sensor Code 48

Figure 4.5 Database for Notifications Sensor 52

Figure 4.6 System Prototype 54

Figure 4.6.1.2 Schematic Diagram 55

Figure 4.6.2 Usability Testing Graph 56

Figure 4.6.3 Functionality Testing Graph 56

Figure 4.6.4 Compatibility Testing Graph 57

Figure 4.6.5 Performance Testing Graph 57

Figure 4.6.6 Recommended Answer 58

Figure 4.6.7 Conclusion Graph 58

**LIST OF APPENDIXISES**

**PAGE**

**APPENDIX I**

Question Section 61

**APPENDIX II**

Poster 65

**APPENDIX III**

Coding I 66

**APPENDIX IV**

Coding II 69

**APPENDIX V**

Prototype Project 70

**APPENDIX VI**

Progress of Project 72

**APPENDIX VII**

Exhibition 74

